



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/864,439	05/24/2001	John R. Applin	10004626-1	2655

22879 7590 05/05/2004

HEWLETT PACKARD COMPANY
P O BOX 272400, 3404 E. HARMONY ROAD
INTELLECTUAL PROPERTY ADMINISTRATION
FORT COLLINS, CO 80527-2400

EXAMINER

KANG, INSUN

ART UNIT	PAPER NUMBER
----------	--------------

2124

DATE MAILED: 05/05/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Application No.

09/864,439

Applicant(s)

APPLIN, JOHN R.

Examiner

Insun Kang

Art Unit

2124

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 24 May 2001 and 28 September 2001.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-20 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-20 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 24 May 2001 is/are: a) ☐ accepted or b) ☒ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date 2.
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____.

DETAILED ACTION

1. This action is responding to application papers dated 5/24/2001 and 9/28/2001.
2. Claims 1-20 are pending in the application.

Drawings

3. The drawings are objected to because:

In the syntax of ***C operator* () const { return *operator -> ();}*** in Fig 2, it appears that & is missing from ***C& operator* () const {return *operator->();}*** as the function returns reference to an object of the class SafePtr.

A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

4. The drawings are objected to under 37 CFR 1.83(a). The drawings must show every feature of the invention specified in the claims. Therefore, the claims 3, 6-8, 10 and 13 must be shown or the feature(s) canceled from the claim(s). No new matter should be entered.

Per claims 6 and 8, Fig 2 shows calling a function which returns as its value a pointer to an object: ***operator->()*** and ***operator* ()*** const calls this function and dereference the pointer to an object to return the reference to an object. It appears to be a typographical error. For the purpose of examination, it is interpreted as "calling a function which returns a pointer to an object."

A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Claim Rejections - 35 USC § 101

5. 35 U.S.C. 101 reads as follows:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

6. Claims 11-16 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Claims 11-16 are non-statutory because they are directed to a "safe pointer class" without recitation of a computer or a computer-readable medium embodying the program. The claims merely recite a "safe pointer class" that is disembodied arrangement so as to be called a "computer program" or compilation of facts, information, or data *per se*, without creating any functional interrelationship, either as part of the stored data or as part of the computing processes performed by the computer ("acts") or computer readable medium so as to enable the computer to perform the claimed steps of checking and processing routine as recited.

Thus the claims represent functional descriptive material that is not capable of producing a useful result, and hence represent only abstract ideas. Therefore, the claims are non-statutory.

Claim Rejections - 35 USC § 112

7. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

8. Claims 1-10, 16 and 17-20 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Per claim 1, it is unclear to what the term "safe pointer type" in the last line of the claim is referring. For the purpose of examination, it is interpreted as "said safe pointer type."

Claim 9 recites the limitation "said" in "said improper native pointer object." There is insufficient antecedent basis for this limitation in the claim. It appears that this claim is dependent on claim 8. For the purpose of examination, it is interpreted that this claim is dependent on claim 8.

Claim 16 recites the limitation "said" in "said improper native pointer object" in line 1 of the claim. There is insufficient antecedent basis for this limitation in the claim.

Per claim 17, the term "likelihood" in claim 17 is a relative term, which renders the claim indefinite. The term "likelihood" is not defined by the claim, the specification does not provide a standard for ascertaining the requisite degree, and one of ordinary skill in the art would not be reasonably apprised of the scope of the invention.

Appropriate correction is required.

As per claims 2-8, 10 and 18-20, these claims are rejected for dependency on the above rejected parent claims 1 and 17.

Claim Rejections - 35 USC § 102

9. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

10. Claims 1-5, 7 and 10-20 are rejected under 35 U.S.C. 102(b) as being anticipated by Pike et al. ("Checkmate: Cornering C++ Dynamic Memory Errors With Checked Pointers," 3/2000) hereinafter referred to as "Pike."

Per claim 1:

Pike anticipates:

-storing and manipulating pointers in a programming language, wherein the programming language supports a native pointer type and standard pointer operations upon a native pointer object of said native pointer type ("checked pointers, a simple wrapper for C++ pointers that prevents pointer arithmetic and other common sources of pointer errors, and detects all ...errors... The syntax of checked pointers is highly faithful to raw C++ pointers, but provides run-time error detection and debugging information," abstract; see also Class Pointer definition in Pointer_2.h of the technical addendum of this article)

-defining a safe pointer type supporting said standard pointer operations (see the class definition of "Pointer" type supporting pointer operations like * (dereferencing operator) and -> (indirection operator) in pg 355 left column);

Art Unit: 2124

-performing automatic pointer checking in association with safe pointer type ("checked pointers...that detects all dereferencing and deallocation errors, including memory leaks," abstract)
as claimed.

Per claim 2:

The rejection of claim 1 is incorporated, and further, Pike discloses checking for a null pointer ("We detect dereferences of null and dangling pointers by recognizing that requested memory addresses are not in this set," pg 354 left column; see also the implementation of "checked pointer," in Pointer_2.h file of the technical addendum of this article) as claimed.

Per claim 3:

The rejection of claim 1 is incorporated, and further, Pike discloses checking for improper pointer alignment ("checked pointers...prevents pointer arithmetic and other common sources of pointer errors, and detects all dereferencing and deallocation errors, including memory leaks, abstract) as claimed.

Per claim 4:

The rejection of claim 1 is incorporated, and further, Pike discloses performing error processing ("detects and reports common bugs such as illegal dereferences," pg 352 left column, Introduction section) as claimed.

Per claim 5:

The rejection of claim 4 is incorporated, and further, Pike discloses that said error processing includes at least one of generating an error message and terminating program execution; generating a warning message without terminating program execution; and invoking a user defined error processing routine ("terminate execution immediately and output a meaningful error message," pg 354 left column; See also operator ->() and operator* () functions in Pointer_2.h of the technical addendum of this article) as claimed.

Per claim 7:

The rejection of claim 1 is incorporated, and further, Pike discloses reading a global variable of said safe pointer type; and performing said standard pointer operation upon said object (the class Pointer is template class, see the class Pointer definition in pg 355 left column) as claimed.

Per claim 10:

The rejection of claim 1 is incorporated, and further, Pike discloses reading a global variable encapsulating an improper native pointer object; and performing error processing in response to said improper native pointer object (the class Pointer is

template class, see the class Pointer definition in pg 355 left column; "keep track of where each pointer is pointing, and which memory locations are currently allocated. We check at run-time that a pointer points to a legal address before deleting or dereferencing it. If the check fails, execution halts and the error is reported immediately," pg 355, left column, paragraph 4) as claimed.

Per claims 11-16, they are the safe pointer class versions of claims 1-5 and 9, respectively, and are rejected for the same reasons set forth in connection with the rejection of claims 1-5 and 9 above.

Per claims 17-20, they are the product versions of claims 1-4, respectively, and are rejected for the same reasons set forth in connection with the rejection of claims 1-4 above.

Claim Rejections - 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

12. Claims 6-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Pike et al. ("Checkmate: Cornering C++ Dynamic Memory Errors With Checked Pointers," 3/2000) hereinafter referred to as "Pike."

Per claim 6:

The rejection of claim 1 is incorporated, and further does not explicitly teach calling a function which returns as its value an object of said safe pointer type.

Official Notice is taken that calling a function with returning statement, within another function is well known in the art of software development, at the time applicant's invention was made. This programming technique is used to make the program easier to understand, modify, avoid redundant coding, reuse, test and debug by breaking a program's task into subtasks, solving these subtasks by sub-algorithms and then calling the subtasks. It would have been obvious for one skilled in the art of computer software development to modify Pike's disclosed T& operator* () function which returns as its value an object of the Pointer class to reuse the T* operator->() by simply calling this function and dereferencing the returned pointer to an object, as this function has a same task of "detect[ing] and report[ing] common bugs such as illegal dereferences (pg 352, left column)" with that in T* operator->(). The modification would be obvious because one skilled in the art would be motivated to reuse the same task without repeating the same code portions in another function.

Per claim 8:

The rejection of claim 1 is incorporated, and further, does not explicitly teach calling a function which returns as its value an object of said safe pointer type. However, see the rejection of the claim 6 above.

Pike further discloses: object encapsulating an improper native pointer object ("checked

pointers, a simple wrapper for C++ pointers that prevents pointer arithmetic and other common sources of pointer errors, and detects all ...errors... The syntax of checked pointers is highly faithful to raw C++ pointers, but provides run-time error detection and debugging information," abstract; see also Class Pointer definition in Pointer_2.h of the technical addendum of this article) and performing error processing in response to said improper native pointer object ("keep track of where each pointer is pointing, and which memory locations are currently allocated. We check at run-time that a pointer points to a legal address before deleting or dereferencing it. If the check fails, execution halts and the error is reported immediately," pg 355, left column, paragraph 4) as claimed.

Per claim 9:

Pike further discloses one of a null pointer, misaligned pointer, and misuse of said pointer in context ("Problems ...are concerned with deleting and dereferencing null and dangling pointers," pg 355, left column, paragraph 4) as claimed.

13. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Insun Kang whose telephone number is 703-305-6465. The examiner can normally be reached on M-F 8:30-5:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on 703-305-9662. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

IK
4/29/2004

Kakali Chak

**KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100**